Document Engineering with TEX and METAPOST

Example Code

Ed Cashin

OIT

Feb. 16, 2000

toc

Contents

ecashin@coe.uga.edu

1

background/test-static.mp

```
% background.mp
%
% MetaPost code for static figure (of form's background)

% the stuff from "verbatimtex" to "etex" is passed to TeX for the
% things in this MetaPost figure that are processed by TeX (usually
% labels).

verbatimtex
\parindent=0in                 % to avoid indentation
\font\twentyssbx=cmssbx10      at 20 true pt
etex

% drawrect is a macro

vardef drawrect(expr llft, urt) =
  draw llft--(xpart llft, ypart urt)--urt--(xpart urt, ypart llft)--cycle;
enddef;

beginfig(1);
  u = 1cm;                     % u is a variable for units
  drawrect((2u,2u),(12u,12u)); % big rectangle
  drawrect((14u,2u),(16u,12u));        % smaller rectangle

  label.rt(btex \twentyssbx STATIC etex, (2u, 13u));

  % do some horizontal lines with a for loop

  for i = 3 upto 11:
    draw (2u, i * u)--(12u, i * u);
  endfor;

  % draw invisible dots to make sure the figure is big enough

  pickup pencircle scaled .01; % microscopic pen
  draw (origin);
  draw (20u, 16u);
endfig;

end
```

2
background/makefile

```
DOC     = test-static
FIGS    = $(DOC).1

$(DOC).pdf : $(DOC).tex $(FIGS)
       pdftex $(DOC).tex < /dev/null
$(DOC).ps : $(DOC).dvi
       dvips -o $(DOC).ps $(DOC)
$(DOC).dvi : $(DOC).tex
       tex $(DOC) < /dev/null

# (you can generate these rules like this)
# perl -e ’for(1..40){print "%.$_\t: %.mp\n\tmpost \$<\n";}’

%.1    : %.mp
       mpost $< < /dev/null
```

3
background/test-static.tex

```
% this tex file is for producing testing documents that have the
% background on them

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\font\fourteenss=cmss10        at 14 true pt
\font\niness=cmss10    at 9 true pt
\font\eightss=cmss10   at 8 true pt
\font\sevenss=cmss10   at 7 true pt
\font\sixss=cmss10     at 6 true pt
\font\fivess=cmss10    at 5 true pt
\font\fourss=cmss10    at 4 true pt
\def\bigspace{\quad\kern -.2em}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\def\thisproject{test-static}

\newdimen\tempdim
\tempdim=11in
\advance\tempdim by -12mm
\hsize \tempdim
\tempdim=8.5in\advance\tempdim by -5mm
\vsize \tempdim
\hoffset -.75in
\voffset -.85in

                               % here we test to see whether we're doing
                               % postscript or pdf:
\ifx\pdfoutput\undefined
       \input epsf
       \special{papersize=11in,8.5in}
\else
       \input supp-pdf                 % mp support for pdftex
       \pdfpageheight=8.5in            % landscape letter pagesize
       \pdfpagewidth=11in
\fi

\nopagenumbers
\parindent=0in

\ifx\pdfoutput\undefined
       \epsfbox{\thisproject.1}
\else
       \convertMPtoPDF {\thisproject.1} {1}{1}
\fi

\bye
```

4

test.pl

```
#! /usr/bin/perl -w
# test.pl
#          typeset args onto PDF form

use strict;
use vars qw($TMP_DIR $ORIG_DIR $PROGBASE);

$TMP_DIR        = "/var/tmp";
$ORIG_DIR       = "/home/ecashin/src/tex/OITdocEngClass/example";
$PROGBASE       = "test";

&process(@ARGV);

sub process {
    my @fields          = @_;
    my $dir             = "$TMP_DIR/$PROGBASE-$$";

    &xsystem("mkdir $dir", "mkdir call");
    my $mp_source       = &make_mp_source($dir, \@fields);
    &run_mpost($dir, $mp_source);
    &symlinks_to_static($dir);
    &run_tex($dir);
    print &slirp_file("$dir/$PROGBASE.pdf");
    &cleanup($dir);
}

sub make_mp_source {
    die "Error: missing parameter" if @_ < 2;
    my ($dir, $fieldsR)         = @_;
    my $basename        = "$PROGBASE-dyn.mp";
    my $filename        = "$dir/$basename";
    open MP, "> $filename"
      or die "Error: could not open $filename for writing";

    #----------backslashes must be escaped when using double quotes
    my $MP_code         = <<"EAT_DAILY";
% $PROGBASE.mp
% MP code generated by $0

verbatimtex
\\parindent=0in                         % to avoid indentation
\\font\\twentyssbx=cmssbx10 at 20 true pt
\\font\\fourteenrm=cmr12 at 14 true pt
etex

beginfig(1);
```

```
  u = 1cm;
EAT_DAILY

    my ($x, $y)                 = (2, 2.5);
    foreach my $field (@$fieldsR) {
       #-------usually you'd have to use a lookup table for finding
       #       the x and y for the field's position, but we use a
       #       simple loop here.

       $field           = &tex_escaped($field);
       $MP_code         .= "  label.rt(btex \\fourteenrm $field etex, "
         . "(${x}u, ${y}u));\n";
       $y               += 1;
    }
    $MP_code            .= <<"WORK_DAILY";

  % use invisible dots at the edges to place the figure
  pickup pencircle scaled .01; % microscopic pen
  draw (origin);
  draw (20u, 16u);
endfig;

end

WORK_DAILY

    print MP $MP_code;
    close MP;

    return $basename;
}

sub run_mpost {
    die "Error: missing parameter" if @_ < 2;
    my ($dir, $mp_source)       = @_;

    my $command         = "cd $dir && mpost $mp_source < /dev/null";

    &xsystem($command, "mpost call");
}

sub symlinks_to_static {
    die "Error: missing parameter" if @_ < 1;
    my ($dir)   = @_;
    my $command         = "cd $dir "
      . " && ln -s $ORIG_DIR/$PROGBASE-static.1 $PROGBASE-static.1 "
       . " && ln -s $ORIG_DIR/$PROGBASE.tex $PROGBASE.tex ";

    &xsystem($command, "symlinking call");
}
```

```
sub run_tex {
    die "Error: missing parameter" if @_ < 1;
    my ($dir)   = @_;

    my $command         = "cd $dir && pdftex $PROGBASE.tex < /dev/null";

    &xsystem($command, "tex call");
}

sub slirp_file {
    die "Error: missing parameter" if @_ < 1;
    my $filename        = shift;
    open SLIRPFILE, $filename
      or die "Error: could not open $filename for reading";
    local $/            = undef;
    my $file_contents   = <SLIRPFILE>;
    close SLIRPFILE;

    return $file_contents;
}

sub cleanup {
    die "Error: missing parameter" if @_ < 1;
    my ($dir)   = @_;
    my $command         = "rm -rf $dir";
    &xsystem($command, "removing $dir call");
}

sub xsystem {
    #--------wrapped system call
    die "Error: missing parameter" if @_ < 2;
    my ($command, $pidgeon)     = @_;
    if (my $error = system $command) {
       $error   /= 256;
       die "Error: $pidgeon exited with abnormal value: $error";
    }
}

sub tex_escaped {
    # NOTE:
    # this subr shows up a lot.
    # this version is current as of Wed Feb 16 11:41:29 EST 2000

    die "Error: missing parameter" if @_ < 1;
    my $text    = shift;
    return "" if (! defined $text) or $text eq "";

    study $text;
    #----------handle quotes
    while ($text =~ s/\"/‘‘/ && $text =~ s/\"/’’/) { }
    #----------escape special chars
    #-----first are extra-special characters: do \, {, and } first
    #-----so that our own TeX code doesn’t get messed up

    # \ is a special case, since using
    # $\backslash$ would conflict with the
    # rule for escaping ’$’
    $text       =~ s/\\/dollabackaslashadolla/g; # \
    $text       =~ s/\$/\\\$/g;                # $
    $text       =~ s/\{/\$\\lbrace\$/g;        # {
    $text       =~ s/\}/\$\\rbrace\$/g;        # }

    $text       =~ s/&/\\\&/g;                 # &
    $text       =~ s/\^/\\char\’136\\relax /g; # ^
    $text       =~ s/~/\\char\’176\\relax /g;  # ~
    $text       =~ s/\|/\$|\$/g;               # |
    $text       =~ s/_/\\\_/g;                 # _
    $text       =~ s/\#/\\char\’043\\relax /g; # #
    $text       =~ s/%/\\\%/g;                 # %
    $text       =~ s/</\$<\$/g;                # <
    $text       =~ s/>/\$>\$/g;                # >
    #---------reset the backslash
    $text       =~ s/dollabackaslashadolla/\$\\backslash\$/g;

    return $text;
}
```

5 test.tex

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% DEFINITIONS
% Ed Cashin's "supersmash" macro, based on smash, TeXbook, p. 360
\def\supersmash#1{\relax\setbox0=\hbox{#1}
       \ht0=0in \dp0=0in \wd0=0in \box0 }

\def\backgroundfig{test-static.1}
\def\foregroundfig{test-dyn.1}

\newdimen\tempdim
\tempdim=11in
\advance\tempdim by -12mm
\hsize \tempdim
\tempdim=8.5in\advance\tempdim by -5mm
\vsize \tempdim
\hoffset -.75in
\voffset -.85in

                               % here we test to see whether we're doing
                               % postscript or pdf:
\ifx\pdfoutput\undefined
       \input epsf
       \special{papersize=11in,8.5in}
\else
       \input supp-pdf                 % mp support for pdftex
       \pdfpageheight=8.5in            % landscape letter pagesize
       \pdfpagewidth=11in
\fi

\nopagenumbers
\parindent=0in

% begin figure inclusion -----------------------------------------------
\ifx\pdfoutput\undefined
       \hbox{\supersmash{\epsfbox{\backgroundfig}}%
           \epsfbox{\foregroundfig}}
\else
       \hbox{\supersmash{\convertMPtoPDF {\backgroundfig} {1}{1}}%
           \convertMPtoPDF {\foregroundfig} {1}{1}}
\fi
\vfil
% end figure inclusion -------------------------------------------------

\bye
```

toc